ARL-TN-13

AR-006-658

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Technical Note 13

A WIND TUNNEL MODEL CONTROL SURFACE ACTUATOR INTERFACE

by

S.A. KENT

Approved for public release.

JUNE 1993

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

AR-006-658

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Technical Note 13

A WIND TUNNEL MODEL CONTROL SURFACE ACTUATOR INTERFACE

by

S.A. KENT

SUMMARY

A microprocessor controlled system is described for the remote positioning of flight control surfaces on a wind tunnel aircraft model. The system utilises DC micromotors and Linear Variable Displacement Transducers (LVDTs) for driving force and accurate position feedback. The Actuator Module was developed primarily for use with a 1/9th scale F/A-18 model to collect data for the International Follow On Structural Test Program (IFOSTP).

POSTAL ADDRESS: **Director, Aeronautical Research Laboratory,**
506 Lorimer Street, Fishermens Bend, 3207
Victoria, Australia.

CONTENTS

1. INTRODUCTION

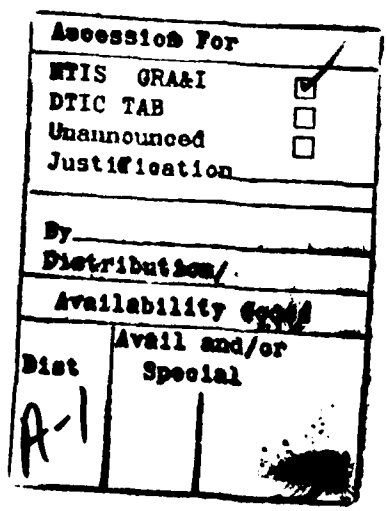
The purpose for the development and construction of the F/A-18 1/9th scale model for the ARL Low Speed Wind Tunnel is to provide data for the International Follow On Structural Test Program (IFOSTP). The F/A-18 aircraft was designed and built for the US Navy as a carrier-based aircraft, and as such the data supplied by McDonnell Douglas need to be expanded to cover the different role the aircraft is to operate in while in service with the RAAF.

The model will be used to study loads on fins and edge extensions, and via flow visualisation will provide data on the vortex burst phenomena which creates dynamic loading on the horizontal stabilator, vertical tailplane and around the engine mounts.

Due to the complex flight envelope of the F/A-18, there is a requirement for the scale model to set flap angle and drive the horizontal stabilators, depending on angle of attack and speed. Actuators have been designed and constructed at ARL to set the angular position of the control surfaces to within ±0.1 degree. These actuators vary in size and consist of DC motors and Linear Variable Displacement Transducers (LVDTs) coupled via a lead screw drive to the control surface (Fig. 1).

The model F/A-18 wing and fuselage support are constructed from Dural 79, a heat treated aluminium alloy, chosen for light weight, strength and surface accuracy when machined. The model is complete with movable trailing edge flaps, ailerons, leading edge extensions and horizontal stabilators. It is the most complex model ever made at the Aeronautical Research Laboratory for testing in the Low Speed Wind Tunnel, and at the time of writing is the only known moving control surface F/A-18 wind tunnel model in the world (Fig. 2).

This actuator technology has been primarily developed for the F/A-18 model. However, it can be used for any subsequent aircraft model, making it a universal system for the future. The actuators are microprocessor controlled and the electronics to operate twelve independent actuators is housed in a module (Fig. 3) that forms a panel of the control console of the ARL Low Speed Wind Tunnel.



2. SYSTEM DESCRIPTION

2.1 Actuator Module Electronics

The Actuator Module (Fig. 4) contains the following hardware:

- CPU card with dedicated Motorola MC68000 microprocessor.
- Memory card with two 2764 (8k x 8) EPROMs and four 6116 (2k x 8) RAM chips.
- Bi-directional Parallel Interface (BPI) card which interfaces the Actuator Module with a master computer.
- Six Actuator cards, each drive the motors and read the corresponding LVDTs for two actuators.
- Relay safety card for computer controlled supply of power to the motors.
- Emergency Stop button (front panel) for power cut-off to the motors in the event of an actuator passing set limits.
- One Gould Econoflex power supply:
  +5 V @ 10 A
  +12 V @ 1.5 A
  -12 V @ 1.5 A
- One ARL designed power supply module:
  +7 V @ 10 A
  +15 V @ 1.5 A
  -15 V @ 1.5 A.

2.2 Model Actuator

The actuators are based on a bell-crank system employing the lever principle of operation for efficient force transmission. A DC micromotor drives a screw thread via a reduction gearbox, which in turn is coupled to the bell-crank. As one end of the crank is driven along the screw thread, the crank rotates about its pivot point, rotating the model control surface. The core of an LVDT is mechanically coupled to another point on the bell-crank. The core is able to move within the windings of the stationary LVDT, producing a sinusoidal signal, the amplitude and phase being proportional to the core distance moved. The Actuator Module decodes the LVDT signal to produce a digital code that represents the precise position of the moving surface.

Although each actuator is based on this system, their size and manner of coupling varies due to space limitations and moving surface type.

The prime considerations in the choice of micromotors were their physical size and torque characteristics. The torque requirements were determined from theoretical wind loading for each of the control surfaces and the physical attributes of the individual gear drive systems.

3. ACTUATOR CALIBRATION

Each actuator is calibrated using the manual control procedure (Section 6.2.5). The leading edge flaps, trailing edge flaps and ailerons are calibrated using metal templates specifically designed and machined for each control surface type and marked in 5 degree increments. The tail stabilisers are calibrated using an inclinometer placed on the control surface.

The actuator to be calibrated is manually driven to the zero degree position and the LVDT reading noted. The surface is then driven to successive angles (usually in 5 degree increments), and the LVDT reading at each angle noted. The host computer uses a mathematical process known as the cubic spline method to produce a smooth interpolation curve passing through all the calibration points. The cubic spline function thus generated converts the LVDT reading to angle and vice versa (Reference 1).

4. ACTUATOR CARD

4.1 Design Considerations

Each Actuator card is a Eurocard of 100 mm x 160 mm, is compatible with the VMEbus and is capable of driving and demodulating two actuators. The cards are controlled by the 68000 microprocessor (Fig. 5 and 6).

Individual address decoding enables the microprocessor to control the motors and reads the LVDT signals of each channel. The motor drive circuitry is capable of bi-directional control of each motor. The LVDT analog signal is decoded and converted to digital binary two's complement format.

All circuitry is interfaced to the microprocessor which is able to receive control commands and send data to a host microcomputer via a sixteen bit bi-directional digital bus (Reference 2).

4.2 Circuit Description

The circuit diagram for a dual channel actuator card is shown in Figure 7. This circuit can be broken down into the following parts:

- Address Decoding
- Motor Drive
- LVDT-to-Digital Conversion
- LVDT Reference Signal.

4.2.1 Address Decoding

Each motor drive latch and LVDT-to-digital converter buffer has its own unique address. The Altera EP900 Erasable Programmable Logic Device (EPLD) is programmed to decode the 23 VMEbus address lines and various control lines to provide a unique chip select signal for each of the buffers and latches. The LVDT-to-digital converter's BUSY signal is also decoded to ensure an LVDT reading is not taken while the converter's output latches are being updated. The EPLD returns DTACK to the VMEbus when a valid bus transfer is initiated. Refer to Appendices 1 & 2 for detailed address decoding information.

4.2.2 Motor Drive

The Sprague UDN-2953 integrated circuit is a full-bridge pulse width modulated motor driver designed for bi-directional control of DC motors with continuous output currents to 2 A. The 7 V supply is switched through the output pins (OUT_A and OUT_B) depending on the logic levels on the OUTPUT ENABLE, PHASE and BRAKE pins.

The truth table for control of the motor drives is as follows:

OUTPUT ENABLE	PHASE	BRAKE	OUT_A	OUT_B	MOTOR FUNCTION
Low	High	High	High	Low	CW drive
Low	Low	High	Low	High	CCW drive
High	X	High	Open	Open	Free-wheel
X	X	Low	High	High	Brake on

The logic levels are latched to the UDN-2953 using a 74LS173 Quad D register integrated circuit.

An Emergency Stop button is located on the front panel of the module, to be used if an actuator should pass set limits. When pushed, it cuts the supply voltage (+7 V) to the motors.

4.2.3 LVDT-to-Digital Conversion

The Analog Devices 2S50 LVDT-to-Digital Converter integrated circuit translates the output from the LVDT into digital form. No external signal conditioning, trims, preamplifiers, demodulators or filters are required.

The 2S50 linearly converts AC signals into an 11-bit parallel digital word. The digital output is an offset binary word which is the ratio between the signal received and reference inputs. The converter is a continuous tracking type and when used with an LVDT, the digital output represents the linear displacement of the transducer.

Being a tracking converter, the output automatically follows the input without the necessity of a convert command. Thus the digital reading must be buffered onto the VMEbus by 74LS645 and 74LS125 buffers.

4.2.4 LVDT Reference Signal

The reference signal used by all the Actuator Module LVDT-to-Digital converters and LVDTs is a 2.5 kHz sine wave generated by the Analog Devices OSC1758 Hybrid Power Oscillator integrated circuit. One oscillator provides the reference signal for all actuator cards. Each card buffers the sine wave through an LM741 operational amplifier to provide the reference signal for two channels.

5. RELAY SAFETY CARD

5.1 Description

The relay safety card is a Eurocard of 100 mm x 160 mm and plugs into the VMEbus backplane. It has a unique address by which the microprocessor can isolate the power supply to the motors and read the status of the relay and Emergency Stop switch (Fig. 8, 9 & 10).

5.2 Address Decoding, Power Control and Status

The Altera EP600 is used to decode address lines A23 to A12 and control lines AS and WRITE to give the relay card the Hex address EDExxx. DTACK is returned to the VMEbus on valid read and write operations.

Power is switched to the motors through a coil and reed relay switch combination (RL1). A write to location EDExxx with Hex data 06 (0110 binary) will turn the relay on while a write to EDExxx with any other data value will turn the relay off.

The status of the relay and Emergency Stop switch can be checked at any time by a read of address EDExxx. The following are possible read-back values:

Hex 00 = Emergency Stop switch open circuit (disabled)
Hex 02 = Safety Relay open circuit (disabled)
Hex 03 = Safety Relay and Emergency Stop switch closed (enabled).

6. MICROPROCESSOR CONTROL

6.1 Software Capabilities

The software is able to:

- Interface with the BPI system.
- Drive up to 12 motors in unison.
- Read up to 12 LVDTs.
- Provide closed loop servo operation for the angular positioning of the model control surfaces.
- Provide manual control of actuator movement.
- Enable software limit setting and control of motor drive to prevent further movement beyond the limits.
- Provide direct LVDT readings.
- Provide status information so the master computer can determine when the actuators have reached their desired target.
- Activate error line and provide error and status codes to reflect fault and running conditions.
- Provide manual control of power to the motors.
- Timeout from main drive loop if a stall condition occurs.

6.2 Software Description

The BPI control routine is written in 68000 assembly language and contains the RAM management statements and all interrupt vector subroutines. The module specific code is written in C and is called from the BPI control routine.

6.2.1 Channel Initialization

When Power-On or a Reset occurs, each channel is sequentially initialized to determine if it is functioning. After an initial LVDT reading is taken, the motor is pulsed UP, then DOWN past the starting position, then UP again. The LVDT reading is taken at each change of direction. The two subsequent LVDT readings are then compared with the original value. If the second reading is higher, and the third reading is lower than the initial reading, the logic status level of that channel is changed to "1", meaning the channel is functional and working correctly. Similarly, if the second reading is lower, and the third reading higher than the initial reading the channel's status is also changed to "1". A non-functioning channel will fail both these tests.

If the test determines that the LVDT readings have not changed correctly, that channel's status is set to "0" and an error code is signalled to the master

computer (Section 5.2.10). The microprocessor will only monitor channels whose status is "1", decreasing motor drive loop time.

Due to construction differences between actuators, the LVDT's and motor's behaviour is not the same for all channels, e.g. a software UP drives the ailerons physically up but the trailing edge flaps down. Likewise the aileron LVDT reading decreases while the trailing edge flap LVDT reading increases.

This requires the software to determine the behaviour of each actuator and monitor the LVDT readings accordingly. Each functioning channel is determined whether it is Type 1 or Type 2, with characteristics as follows:

Type 1: software UP for LVDT reading increase
software DOWN for LVDT reading decrease

Type 2: software UP for LVDT reading decrease
software DOWN for LVDT reading increase.

The channel initialization procedure loads each actuator type into memory, to be checked when driving the motors.

The program then enters a wait loop which tests for changes in limits and acts on received movement trigger and manual test commands from the host computer.

6.2.2 BPI Interface

When the host computer addresses the Actuator Module an interrupt is initiated on IRQ5 of the VMEbus. The processor handles the interrupt by branching to a subroutine determined by the eight bit vector received. All addresses are in the form of 16 bit words where an even address (e.g. 8B60) is decoded as a FETCH (read) cycle, and an odd address (e.g. 8B61) is decoded as a PASS (write) cycle. For specific technical information on the BPI interface refer to Reference 2.

The address is formed by two eight bit bytes, the upper byte being the module specific address while the lower byte is a vector pointer. The vector pointers enable 255 individual software routines to be selected.

The specific address for the actuator module has been allocated as "8BXX", where the vector pointer XX in Hex code is selected from the decimal range 00 to 255. Vector00 to vector63 are for system use only while vector64 to vector95 are assigned to user interrupts, which are used by VMEbus cards other than the actuator controller. The actuator controller card uses read/write vectors starting at vector96 (8B60) and continuing through to vector255 (8BFF). The first sixteen vectors are used as error vectors, module identification string and clearing error/status locations. The main movement trigger vector (8B69) is also included in this group.

The twelve actuator control channels have each been allocated four read vectors and four write vectors. This accounts for vector112 through to vector207.

Specific data fetched from the actuator controllers by the master computer include:

- LVDT readings
- Upper limit LVDT settings
- Lower limit LVDT settings.

while data passed to the actuator controller by the master computer include:

- Target LVDT readings
- Upper limit LVDT setting
- Lower limit LVDT setting
- Simultaneous start for all motors (Trigger command).

Various vectors above vector207 are used for motor status, manual test mode and LVDT read-only function.

Appendix 3 contains detailed BPI vector information.

6.2.3 Motor Drive and LVDT Reading

The VMEbus address for each motor is held in a static array. A pointer to the motor's address is set up and assigned the direction to move (UP, DOWN or STOP). This places the direction logic on the VMEbus which is then clocked into the motor drive latches when the correct motor address is placed on the VMEbus. Because the data are latched to the motor drive circuit, the move command need only be sent once, and the motor will continue to move until the STOP command is received.

As for the motor drive, each LVDT's VMEbus address is held in a static array which is accessed at a certain position depending on the LVDT to be read. A pointer is set up to read the *contents* of the LVDT address (the LVDT reading itself) and the acquired value is assigned to a variable. As the LVDT reading is an 11 bit word and the variable is 16 bits long, the LVDT reading must be logically ANDED with 07FF Hex (2047 decimal) to zero the state of the five most significant bits.

6.2.4 Control Surface Positioning

A data write to BPI vector 8B69 will initiate the control surface movement procedure. The LVDT target (reading to move to) variable of each channel with status "1" (i.e. functioning) is then read. If the target has changed since the last read, that channel's status is changed to "2" to reflect that the channel is required to move.

The LVDT position of every channel with status "2" is taken and the reading compared with the target position. If the current reading is lower than the target, the motor is directed UP and that channel's status changed to "3". If the reading is higher than the target, the motor is directed DOWN and status changed to "4".

A loop is then entered in which the LVDT readings of channels with status "3" or "4" are continually monitored. If a channel reaches an upper or lower limit, the motor is stopped, pulsed back off the limit slightly, and its status reset to "1". Providing no limit is encountered, the LVDT reading will reach a preset cut-off value before the target value, causing the motor to pulse until the target is obtained. That channel's status is then reset to "1".

The loop is continued until the status of all channels requiring movement has returned to "1". The program then reverts back to the main wait loop.

6.2.5 Manual Control

A case may exist, particularly during actuator calibration, where a specific channel is required to be manually driven. This means the motor can be directly controlled by the master, i.e. UP, DOWN or STOP without the need for target angles to be entered. In this mode only the LVDT reading is available to be read by the master computer.

The manual test routine is initiated with a data pass to BPI address 8BD1, followed by a data value which equals the channel number to be driven.

e.g. a data pass to 8BD1 with data 0003 initiates testing of channel 3.

This clears a memory location tested for "0" by the main code. On a successful test the main program branches to the manual drive routine.

The motor is started with a data pass to BPI address 8BD3, followed by a data value for direction:

0001 = UP (clock-wise)
0002 = DOWN (counter clock-wise)
0003 = STOP.

To exit the routine a pass cycle to BPI address 8BD5 is executed. No specific data are required to be sent with this vector write, but data must be included to successfully complete the pass. This clears a flag in another memory location which causes the program to drop out of the manual drive routine. The manual test mode for a particular channel must be exited before another channel can be tested.

During the test procedure the raw LVDT reading is returned to that particular channel's LVDT reading address.

For example: A pass to address 8BD1 followed by data word 0005 initiates testing of channel 5. A pass to address 8BD3 followed by data word 0002 drives the motor DOWN. A subsequent pass to 8BD3 followed by 0003 STOPs the motor. A last pass to address 8BD5 followed by any data value exits the test routine.

For precise positioning of the actuators, a "pulse" mode can be entered where the motors will move in short pulses in the direction specified. Pulse mode can be turned on by a pass to address 8BDB and turned off by a pass to address 8BDD. No specific data are required to be sent.

6.2.6 Software Limits

At the start of the program the master limits are set to zero. Part of the wait loop checks all input limit BPI vectors (both upper and lower) and compares the value with the current limit setting. If any channel's limit vectors have changed a status check is performed. If a non-functioning channel has been selected an error is signalled to the master computer. On selection of a functioning channel, the new limit value is stored. If an attempt is made to drive a channel with no specified limits, an error code is sent to the master computer. A channel cannot be driven without limits being specified.

During the motor movement phase the LVDT readings are continually compared with the current limits, i.e. if the motor is moving counter clock-wise, the LVDT reading is checked against the lower limit, while the upper limit is checked for a motor moving clock-wise. If an LVDT reading exceeds a limit, motion is stopped and an error code sent to the master computer. The motor is then momentarily pulsed in the reverse direction to clear the limit, and the channel's status is returned to "1".

6.2.7 LVDT Reading Display

The LVDT readings for each channel are available to be read by the master computer at any time. After the actuator initialization process the LVDT readings for every channel are taken and stored in their respective BPI vector addresses. During the motor movement phase the LVDT readings for all functioning channels are constantly updated.

Bit 13 (for channels 1 to 4), bit 14 (channels 5 to 8) or bit 15 (channels 9 to 12) is set to "1" in the BPI status register, informing the master computer that valid data are available. A pointer to the data buffer is set and incremented by an offset determined by the channel number. The LVDT reading is then stored in this location. The master computer accesses these values by addressing the appropriate BPI vectors.

6.2.8 Status Information

A BPI status register informs the master computer of the current movement status of each actuator. A fetch of BPI vector 8BD0 is required to access this register.

The lower 12 bits (bit0 - bit11) are used as status flags. Bit 0 represents Channel 1, bit 1 Channel 2, bit 2 Channel 3 and so on to bit 11 representing channel 12. If a bit is set to a "1" the motor is currently moving, a "0" means the motor is stationary. A test of these flags will determine if a motor is moving or not.

The three most significant bits (13 - 15) are always "1", so the word read from address 8BD0 will start with "E".

The correspondence between status bits and status is as follows:

Bit pattern	Hex	Status
1110000000000001	E001	Channel 1 moving
1110000000000010	E002	Channel 2 moving
1110000000000100	E004	Channel 3 moving
1110000000001000	E008	Channel 4 moving
1110000000010000	E010	Channel 5 moving
1110000000100000	E020	Channel 6 moving
1110000001000000	E040	Channel 7 moving
1110000010000000	E080	Channel 8 moving
1110000100000000	E100	Channel 9 moving
1110001000000000	E200	Channel 10 moving
1110010000000000	E400	Channel 11 moving
1110100000000000	E800	Channel 12 moving

If several motors are moving simultaneously, their respective bits will be set to a "1" to reflect their status:

Bit pattern	Hex	Status
1110000101001101	E14D	Channels 1,3,4,7,9 are currently moving

If all bits are "0", then all motors are stationary. This results in the Hex word E000 being read from address 8BD0.

6.2.9 Error and Status Codes

If a "fault" condition occurs, the module signals the master computer by asserting the BPI error line and placing an error code in BPI vector96 (address 8B60). See Reference 1 for detailed information on the BPI error process.

The value of the error code depends on the detected fault and identified channel. The following error codes are specific to the Actuator Module:

EAx0 - actuator not functioning
EAx1 - master set upper limit reached
EAx2 - master set lower limit reached
EAx3 - upper limit not specified
EAx4 - lower limit not specified
EAx5 - timeout expired - actuator stalled

EAF0 - Emergency Stop switch has been disabled
EAF1 - Motor power safety relay has been disabled.

The "x" is substituted by the Actuator Channel number that is associated with the error code.

Valid Actuator Channel numbers are a single Hex value in the range "1" to "C".

e.g. EA10 denotes Channel 1 actuator not functioning,
EA51 denotes Channel 5 master set lower limit reached,
EAB4 denotes lower limit for Channel 11 has not been set,
- and so on.

The error codes are handled in a string fashion. If a code is present in the error code storage location and has not been fetched by the master computer, the next error is placed in the next memory location. Thus, if the error flag is asserted, a continuous fetch from address 8B60 is required until the NULL is detected. Reading the NULL means that no more errors are present (at that time) and the master computer must execute a data pass to addresses 8B65 and 8B67 to clear the error status buffers and error flag.

The advantage of this system is that if a number of errors occur in a short time frame, the corresponding codes will not be missed by the master computer while it is processing information or errors from other modules.

6.2.10 Motor Power Control

The status of the Emergency Stop switch and motor power relay are continuously monitored. If one or the other are found to be open circuit (disabled) an error code is signalled to the master computer. If a power error

condition exists the error code is only signalled once, to prevent constant errors being read by the master computer. The error code routine is re-enabled when power is restored to the motors.

The motor power relay can also be controlled manually through the BPI. A data pass to address 8BD9 followed by data word 0006 will turn the relay on while a pass to 8BD9 followed by any data other than 0006 will turn the relay off.

6.2.11 Drive Loop Timeout

A timeout feature is included in the main actuator drive loop. If a working actuator is commanded to move but stalls during the movement phase or does not move at all due to mechanical or component failure, the drive loop is exited after a period of approximately thirty seconds, and a timeout error code sent to the master computer.

The timeout is implemented by decrementing a variable on each pass through the drive loop. If a channel's count variable reaches zero, that channel's status is reset and its timeout error code sent.

7. MODULE SOFTWARE TEST PROCEDURE

7.1 Initial Test Configuration

7.1.1 Hardware

The test configuration consists of the following hardware:

- 1 IBM compatible personal computer
- 1 BPI test interface unit
- 2 CK600 video display units
- 1 Safety interface circuit
- 2 Actuator test rigs. (Fig. 11).

The Actuator Module (Fig. 12) consists of the following cards:

- Slot 1: M68000 CPU card
- Slot 2: Memory card - 8 x 6264 static RAM (base address switchable from 10000H to 00000H)
- Slot 3: Memory card - 2 x 2732 EPROMs containing monitor program MONTOR V11.1, 2 x 6116 static RAM (base address switchable from 00000H to 10000H)
- Slot 4: BPI card
- Slot 5: Asynchronous communications card (ACIA)
- Slots 6-11: Actuator cards.

7.2 Software - Test Version

The compiler software used for compiling and testing the Actuator Module code is the "HI-TECH C" C to MC68000 cross compiler.

The specific Actuator Module test code (ACTUTEST.C) is written in C language using a basic text editor and is embedded in the BPI control code (FA18TEST.AS) written in MC68000 machine code. After either file has been edited, it is necessary to recompile the complete code by running the batch file MKACTTST.BAT. This executes the MAKE utility which processes a file called ACTUTEST.MAK. This links all relevant library and header files with the original code and creates FA18TEST.OBJ and FA18TEST.HEX.

7.2.1 Testing Software using the 68K Monitor

Before the .HEX file can be downloaded to the module, the serial port COM1 must be configured to 4800 baud to match the baud rate set in the ACIA card. This is done by executing the DOS MODE command in the following manner:

MODE COM1:4800,n,8,1 <return>

The lead from the ACIA card is plugged into the COM1 port and the memory select switch is in the "MON" (monitor) position. The Monitor prompt "Hi!" should appear on the CK600 display unit (Module terminal) connected to the ACIA card.

At the DOS prompt type:

CTTY COM1 <return>

then type:

H <return>

on the Module terminal to gain access to the personal computer. The DOS prompt should appear on the Module terminal screen. The .HEX file is then downloaded by running D.BAT. On the Module terminal keyboard type:

D FA18TEST <return>

The message "starting dump fa18test.o" should appear on the screen. When downloading of the file is complete the "Hi!" prompt reappears on the Module terminal screen. Control is restored to the P.C. by typing:

H <return>

then type:

CTTY CON <return>

on the Module terminal. Typing the key combination Ctrl-C on the Module terminal then restores the monitor prompt "Hi!".

The test code is compiled to download to address 10000H in RAM but is linked to start execution at 00000H. The program is therefore run by pressing and holding the RESET button on the CPU card and switching the memory select switch to the "RAM" (RAM memory) position. This switches the base address of the RAM memory card from 10000H to 00000H, and the base address of the card containing the monitor EPROMs from 00000H to 10000H. Releasing the RESET button then starts the downloaded program.

Test messages should appear on the Module terminal while BPI read and write functions are carried out on the CK600 terminal connected to the BPI interface unit.

7.2.2 Testing Software Using ROM Emulators

The ROM emulator units allow the .HEX file to be downloaded directly to the memory card in parallel format, through LPT1, and replace the EPROMs that usually reside in the memory card.

The batch file ROM.BAT is used to split the .HEX file into separate upper and lower byte files and direct the two files to the ROM emulators. The program BPIMON.EXE is then executed to allow communication to the module via the BPI card installed in the P.C.

To download the Actuator Module software type:

ROM FA18BPI <return>

7.2.3 Software - Final Version

The final (EPROM) version of the Actuator Module code is edited and compiled in a similar manner to the Test version, although different files are used. The final version of Actuator Module specific code is called ACTUATOR.C and is embedded in FA18BPI.AS. These two files are compiled using the batch file MKACTU.BAT. This batch file executes the MAKE utility which processes ACTUATOR.MAK. The resultant .HEX file is FA18BPI.HEX.

The hex file FA18BPI.HEX is then post-processed as detailed in the next section and the resultant .OBJ file used to program the EPROMs.

7.2.4 ROM Procedure

Enter the C:\UNI-PROG directory on the computer with the Universal Programmer connected to it.

To create a binary file that the EPROM programmer software can load type:

HEXOBJ2 [drive]FA18BPI.HEX <return>

where **[drive]** is the disk drive with the file on it (e.g. **A:**).

Accept the default output filename (FA18BPI.BIN), then type M for Motorola format. Use the start address default (00000000).

To run the programmer software type:

EPP02 <return>

and load the binary file **FA18BPI.BIN** with option 2. Enter 0 for buffer starting address. Use "M" and "T" to select EPROM manufacturer and type, then "P" to program the EPROMs.

When in the Program mode type "O" to program the odd (upper) byte, then "E" to program the even (lower) byte. When programming four 2764s, after the first two (odd and even) EPROMs adjust the "Buffer Starting Address" by typing "C", entering "0000" for the Chip Start Address, "1FFF" for the Chip End Address, and entering "4000" as the Buffer Start Address. Program the final two EPROMS (even and odd) as for the first two.

Refer to Fig. 13 for odd and even byte EPROM positioning.

8. MODULE HARDWARE

8.1 Actuator Module

The Actuator Module is comprised of a standard 19 inch rack mountable single height chassis. All plug-in cards are accessed from the rear of the module. The microprocessor backplane contains 13 slots and is VMEbus compatible. All cards are constructed to Eurocard standards and the module hardware is built from standard Eurocard components.

8.2 Interface Cabling

Cabling from the Actuator Module to the model is achieved with 10 metres of shielded cable to the Actuator Module Terminal Box mounted on the Wind Tunnel, then eight metres of super flexible cables from the Terminal Box to the model (Fig. 14).

CONCLUSION

The system was thoroughly evaluated within the laboratory prior to installation in the ARL Low Speed Wind Tunnel where it performed reliably throughout the duration of the tunnel test program. Many thousands of readings were taken over a wind speed range of 0 to 50 metres/second. All control surfaces were rapidly and accurately positioned on command from the host computer. The Actuator Module should be suitable for control applications associated with most future requirements.

ACKNOWLEDGMENTS

The author would like to thank J.F. Harvey and O.F. Holland for their contributions during the development of the hardware and software for the Actuator Module.

REFERENCES

1. S.S.W. Lam & Y.Y. Link F/A-18 Actuator Control Software Program. ARL Technical Note ##, To be published.

2. J.F. Harvey A Data Acquisition Parallel Bus for Wind Tunnels at ARL. ARL Flight Mechanics Technical Memorandum 412, August 1989.

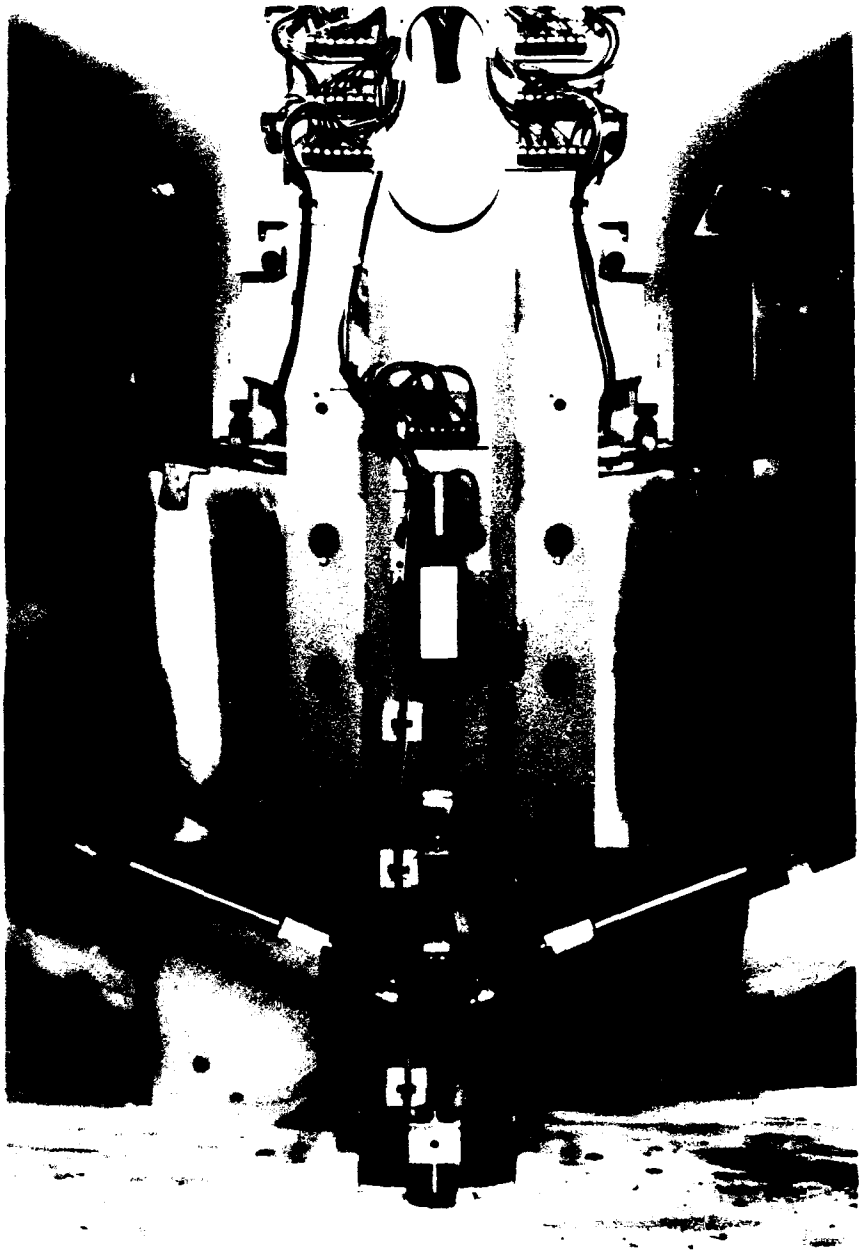


Figure 1: Leading Edge Flap Actuator

Figure 2: Complete Model in ARL Low Speed Wind Tunnel

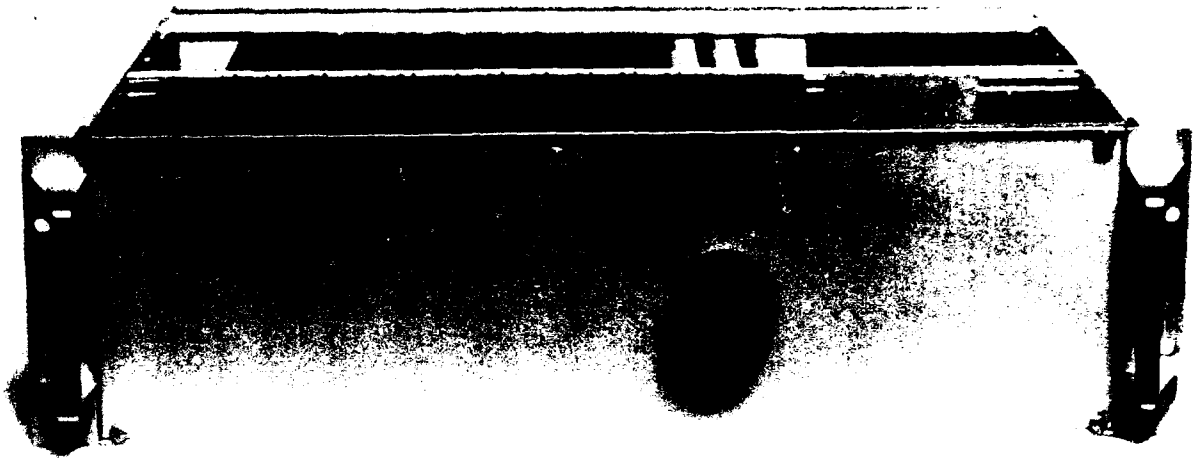


Figure 3a: Front view of Actuator Module

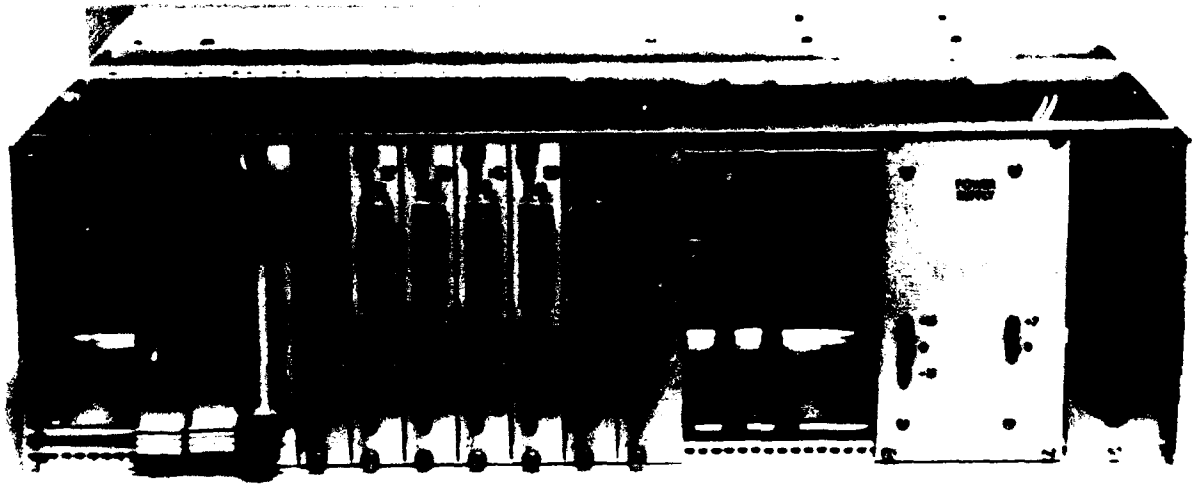


Figure 3b: Rear View of Actuator Module

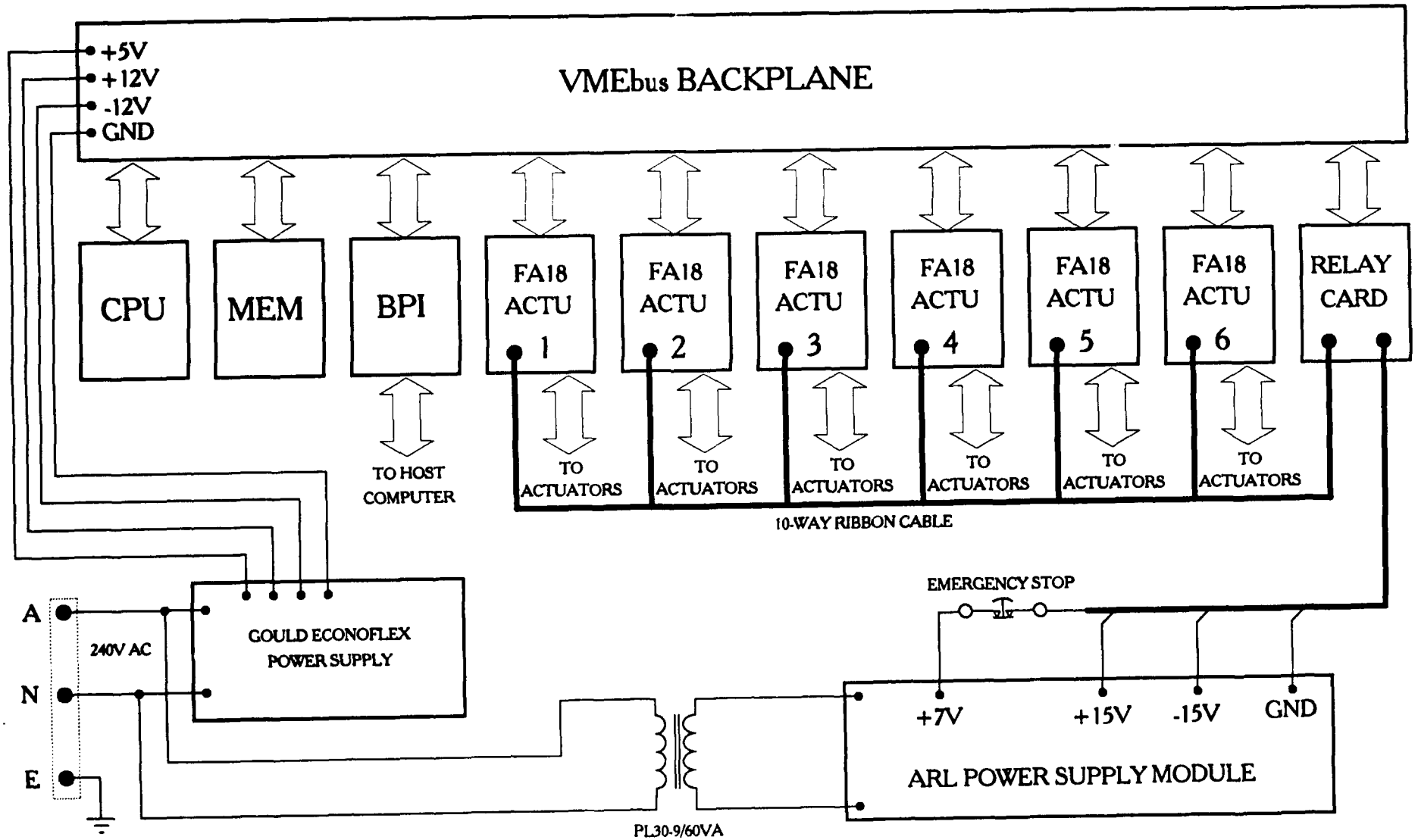


FIGURE 4: ACTUATOR MODULE BLOCK DIAGRAM

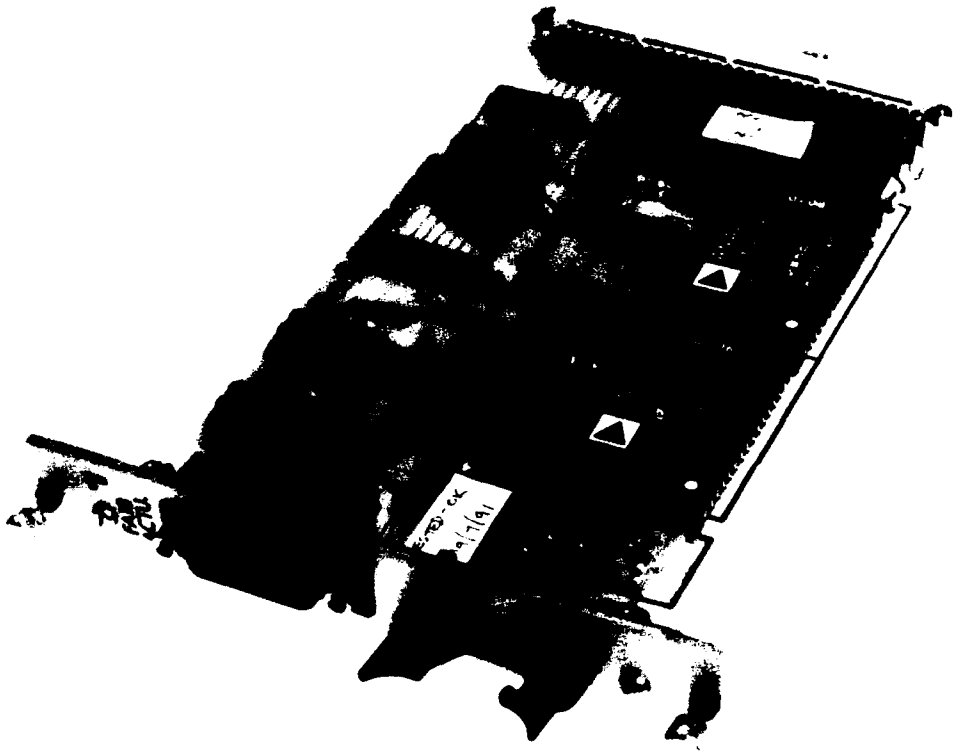


Figure 5: Actuator Card

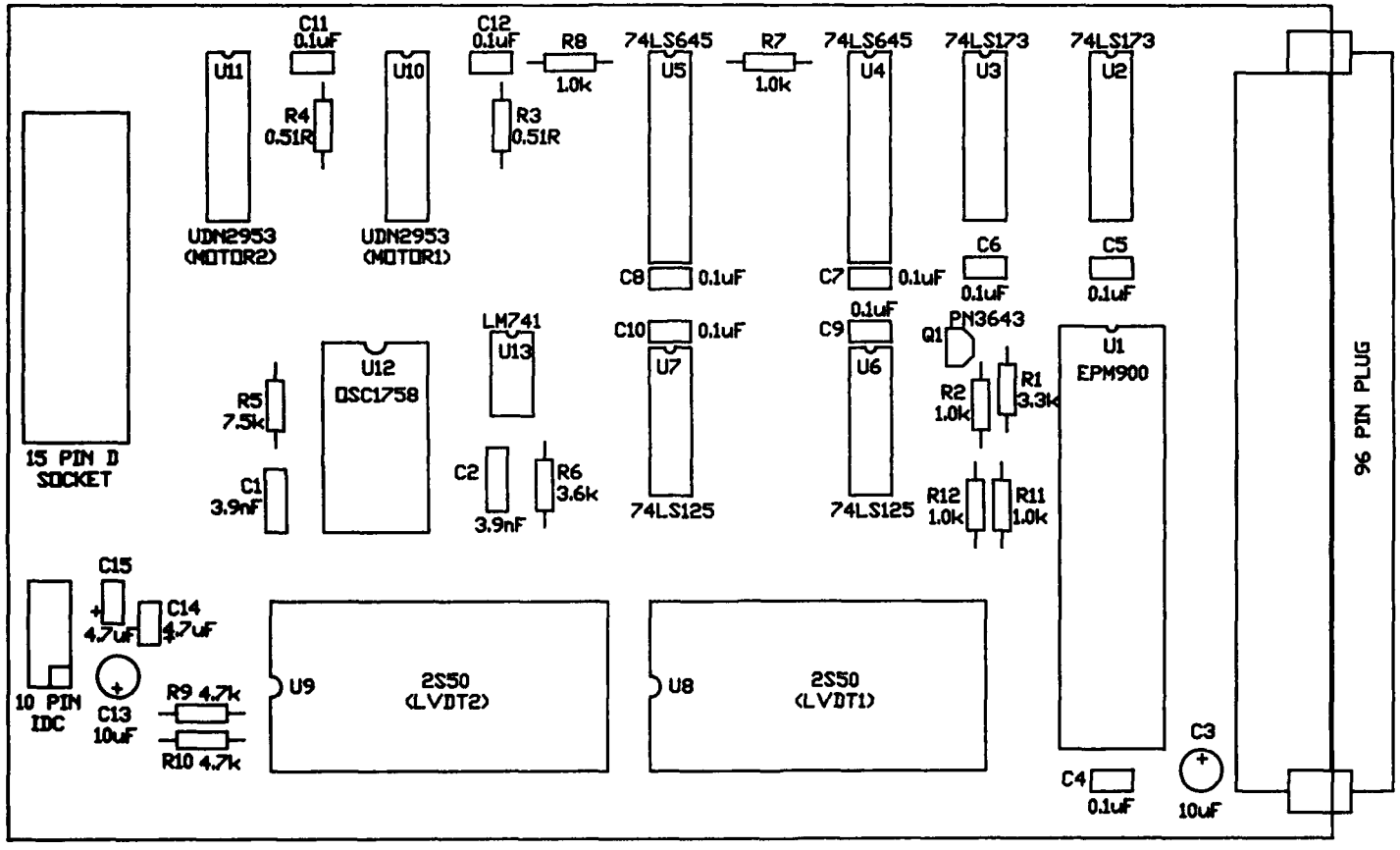
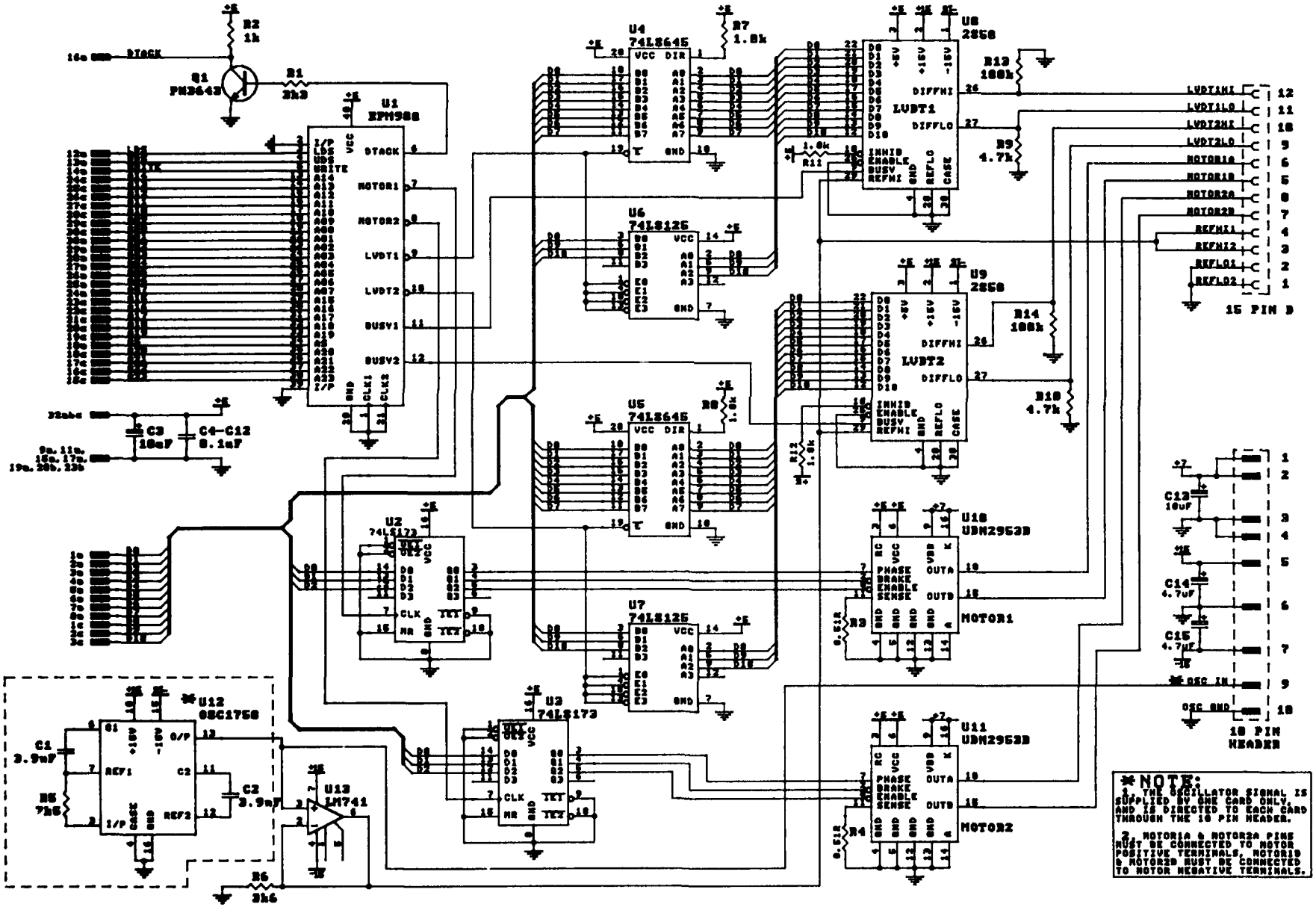


FIGURE 6: ACTUATOR CARD COMPONENT OVERLAY



NOTE:

1. THE OSCILLATOR SIGNAL IS SUPPLIED BY ONE CARD ONLY, AND IS DIRECTED TO EACH CARD THROUGH THE 18 PIN HEADER.

2. MOTOR1A & MOTOR2A PINS MUST BE CONNECTED TO MOTOR POSITIVE TERMINALS, MOTOR1B & MOTOR2B MUST BE CONNECTED TO MOTOR NEGATIVE TERMINALS.

FIGURE 7: ACTUATOR CARD CIRCUIT DIAGRAM

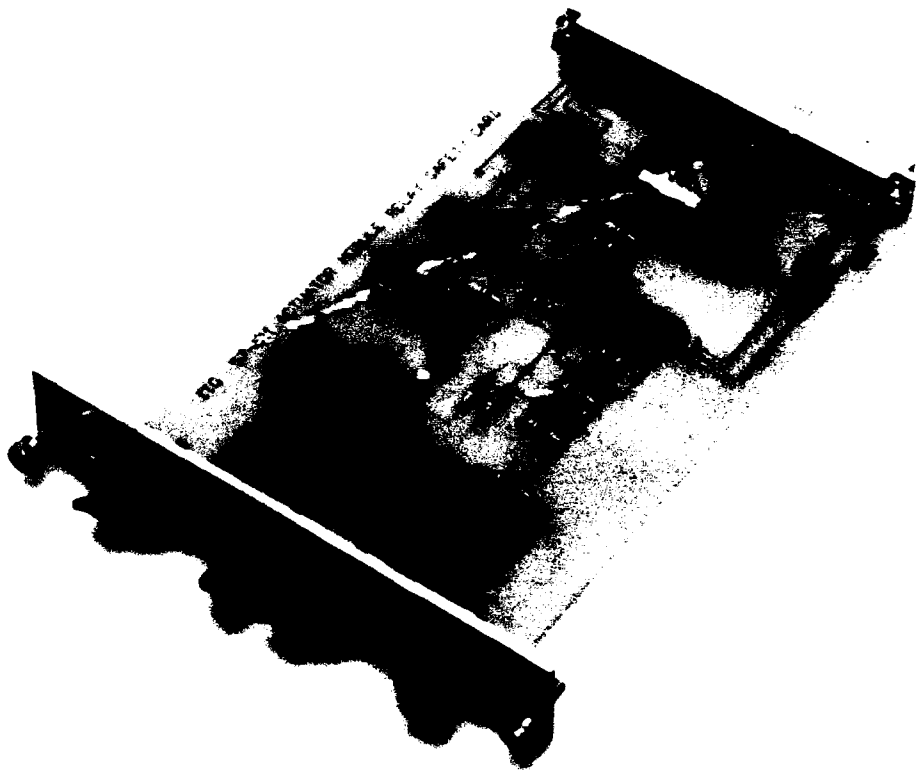


Figure 8: Relay Safety Card

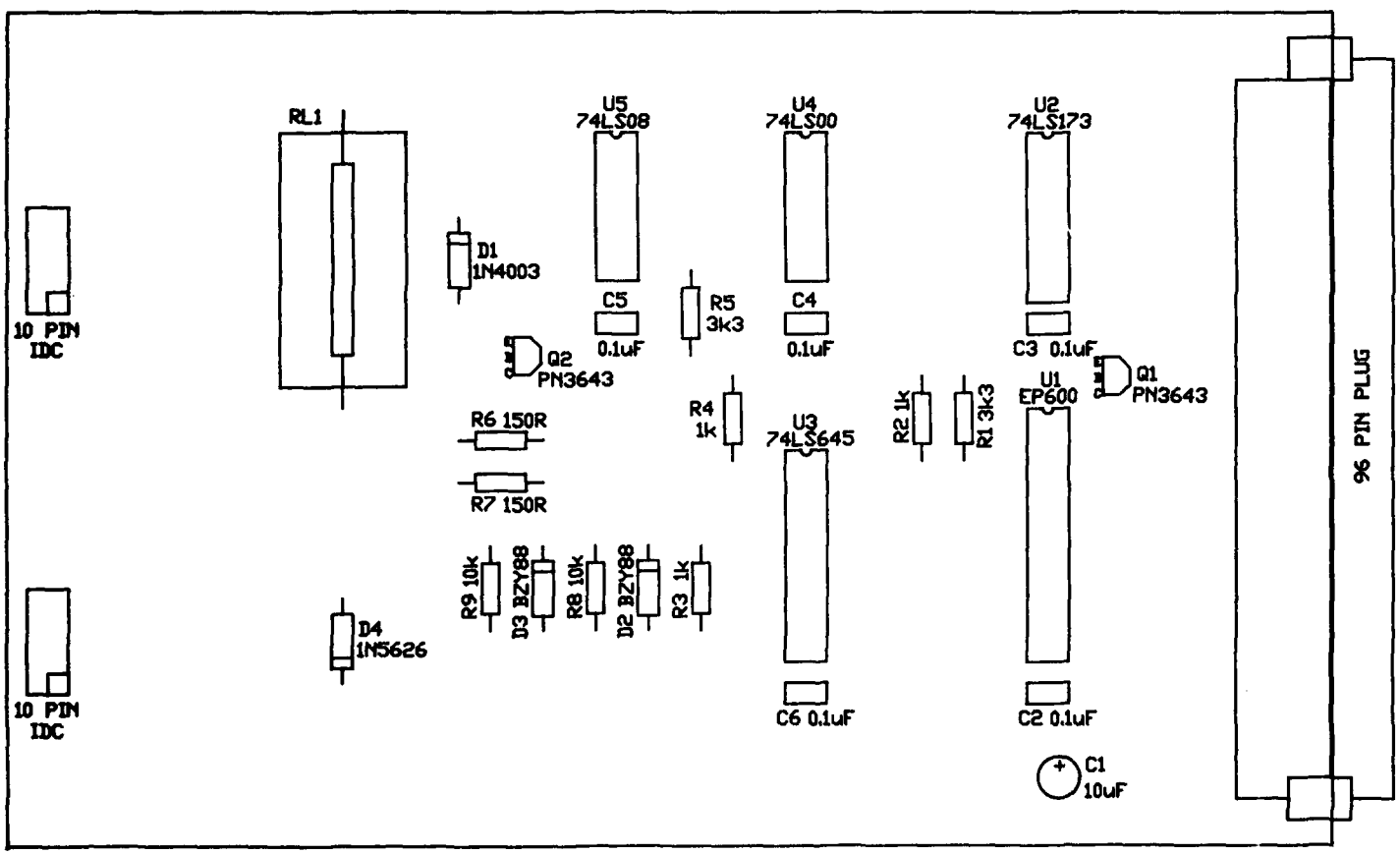


FIGURE 9: RELAY SAFETY CARD COMPONENT OVERLAY

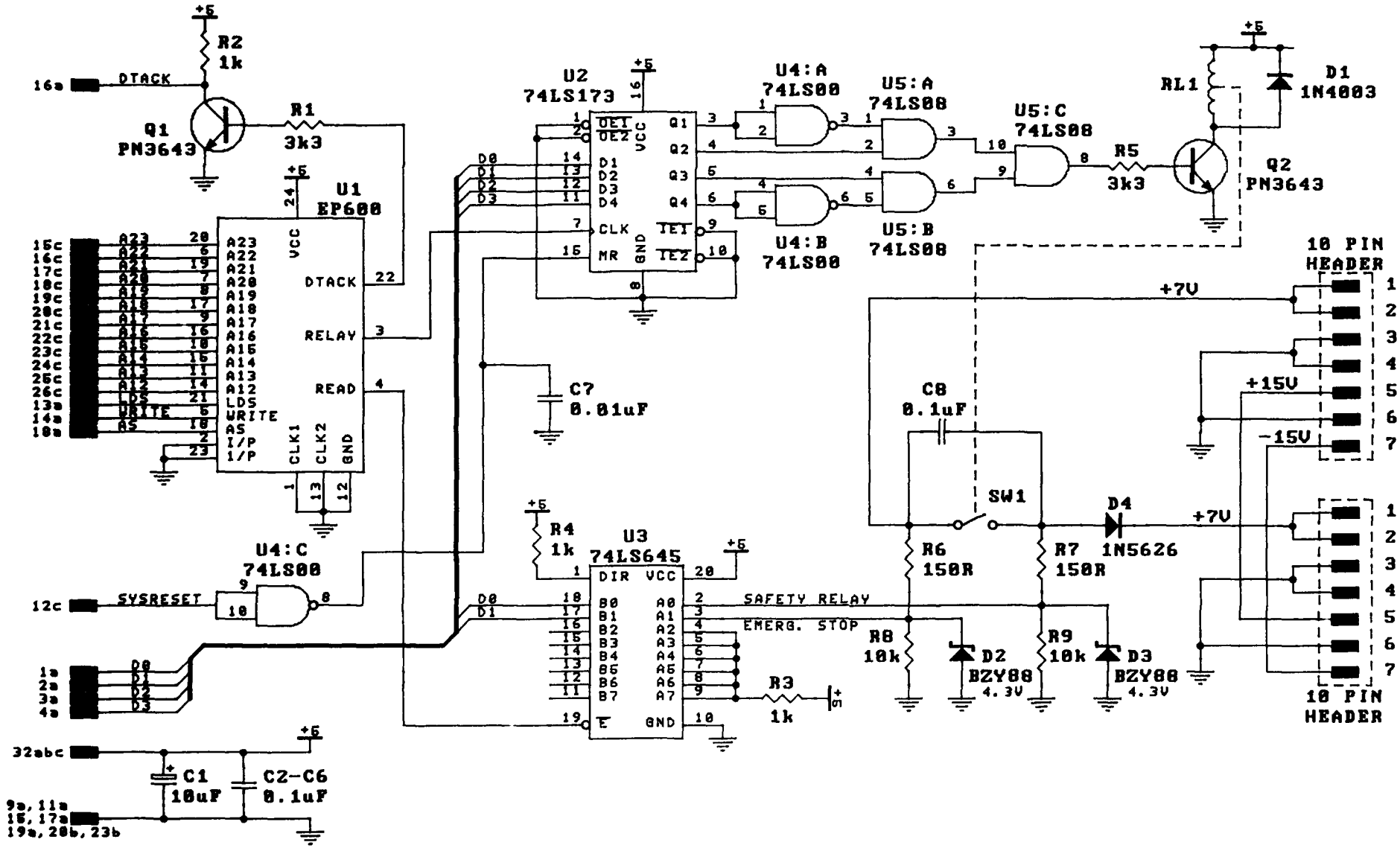


FIGURE 10: RELAY SAFETY CARD CIRCUIT DIAGRAM

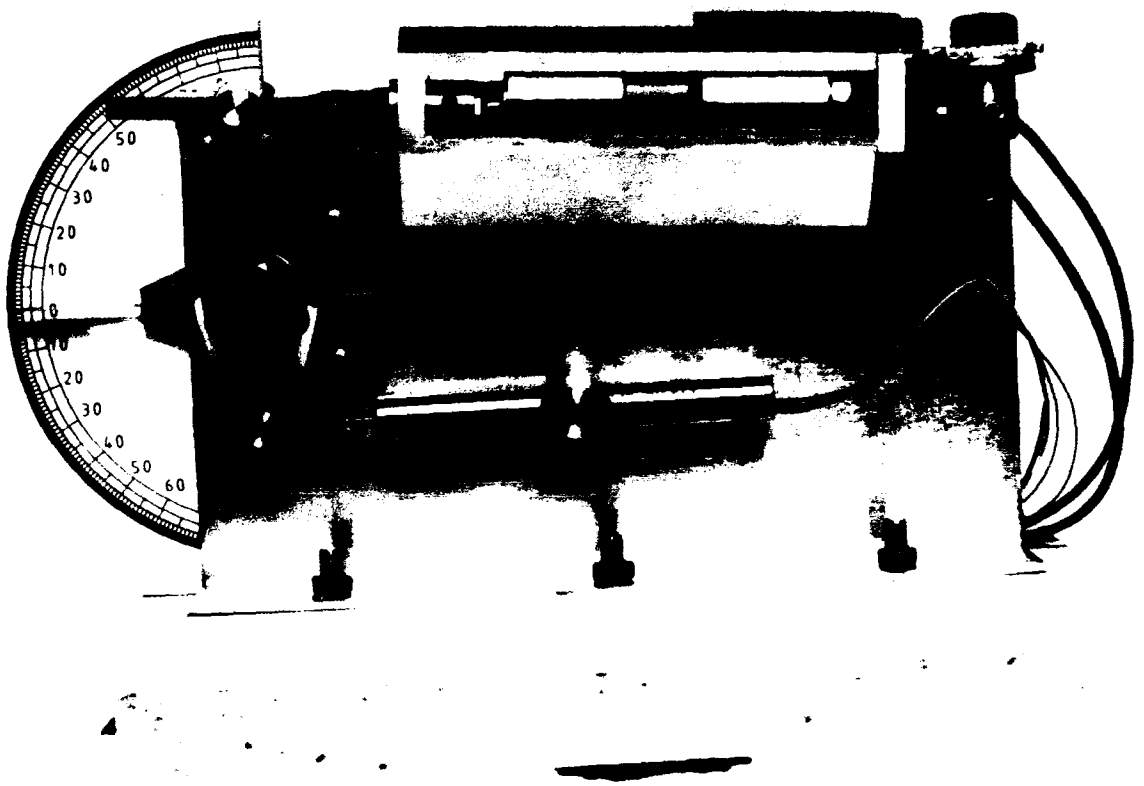


Figure 11: Actuator Test Rig

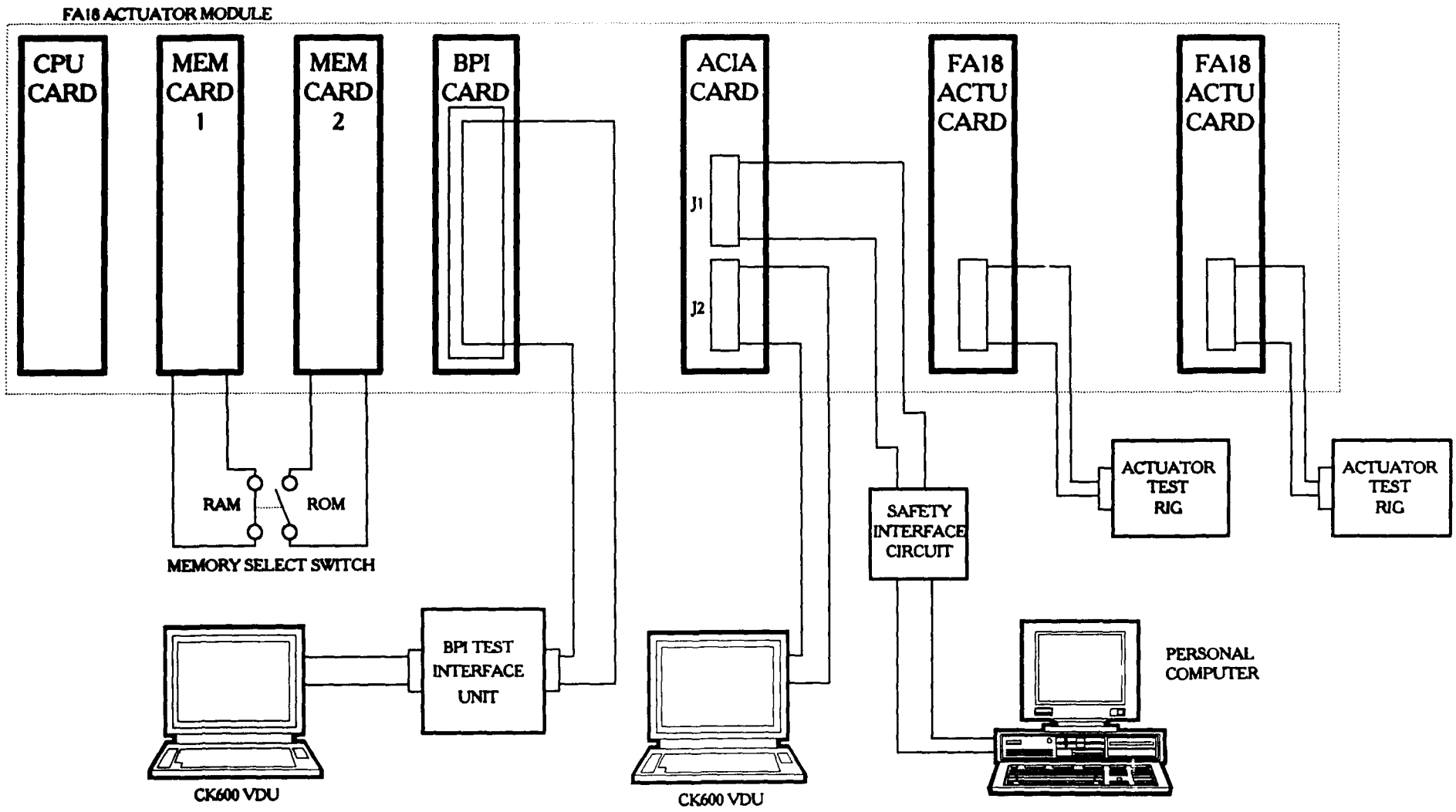


FIGURE 12: ACTUATOR MODULE TEST CONFIGURATION

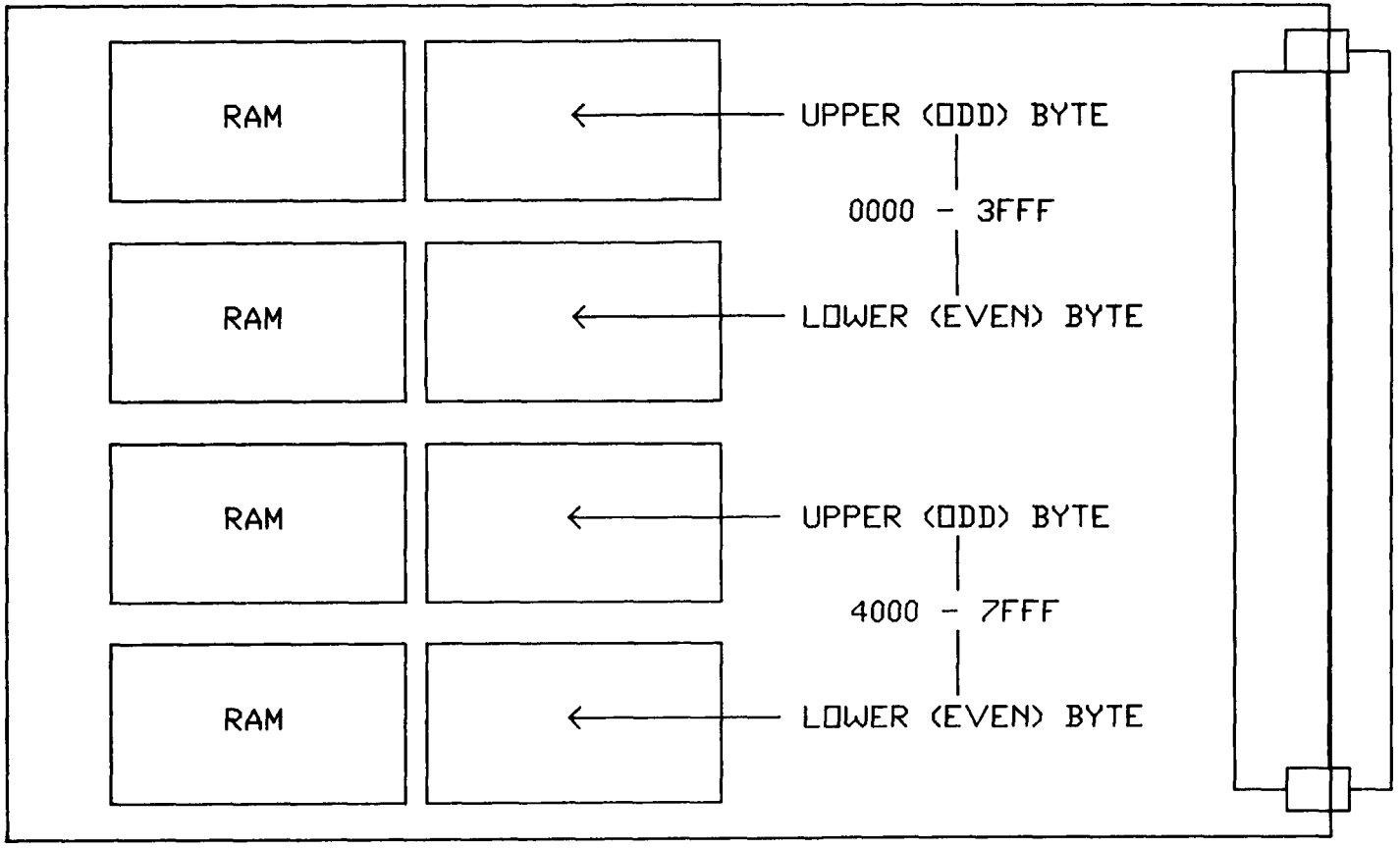


FIGURE 13: MEMORY CARD EPROM POSITIONS

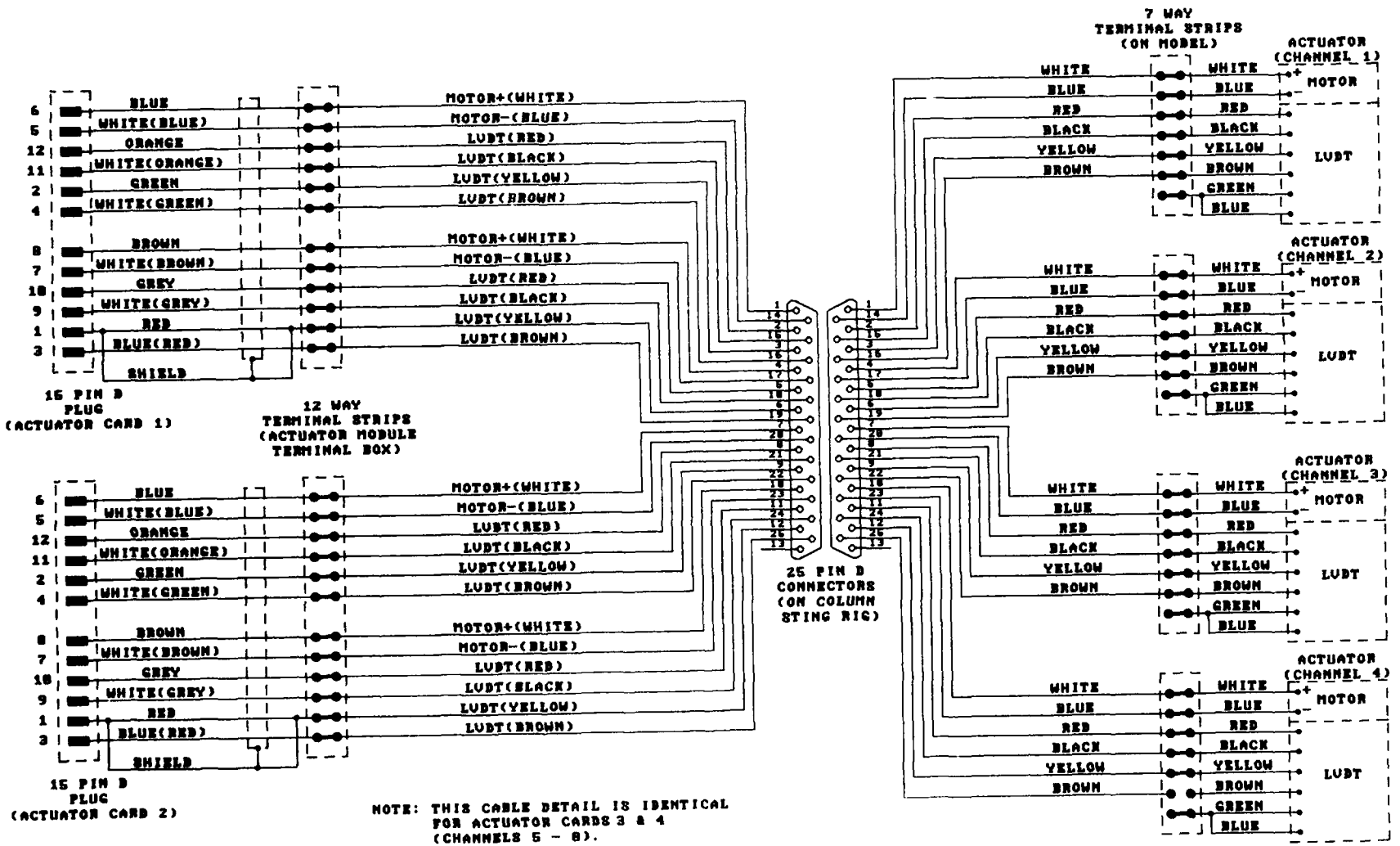


FIGURE 14: INTERFACE CABLE DETAILS FOR F/A-18 MODEL

APPENDIX 1

ADDRESS DECODING FOR MODEL ACTUATOR CARDS

DECODER CHIP: ALTERA EP900DC EPLD

CARD	DEVICE	ADDRESS	ADDRESS LINES: A23...AS
1	MOTOR1	FFD010	1111 1111 1101 0000 0001 0000
	LVDT1	FFD012	1111 1111 1101 0000 0001 0010
	MOTOR2	FFD014	1111 1111 1101 0000 0001 0100
	LVDT2	FFD018	1111 1111 1101 0000 0001 1000
2	MOTOR1	FFD020	1111 1111 1101 0000 0010 0000
	LVDT1	FFD022	1111 1111 1101 0000 0010 0010
	MOTOR2	FFD024	1111 1111 1101 0000 0010 0100
	LVDT2	FFD028	1111 1111 1101 0000 0010 1000
3	MOTOR1	FFD030	1111 1111 1101 0000 0011 0000
	LVDT1	FFD032	1111 1111 1101 0000 0011 0010
	MOTOR2	FFD034	1111 1111 1101 0000 0011 0100
	LVDT2	FFD038	1111 1111 1101 0000 0011 1000
4	MOTOR1	FFD040	1111 1111 1101 0000 0100 0000
	LVDT1	FFD042	1111 1111 1101 0000 0100 0010
	MOTOR2	FFD044	1111 1111 1101 0000 0100 0100
	LVDT2	FFD048	1111 1111 1101 0000 0100 1000
5	MOTOR1	FFD050	1111 1111 1101 0000 0101 0000
	LVDT1	FFD052	1111 1111 1101 0000 0101 0010
	MOTOR2	FFD054	1111 1111 1101 0000 0101 0100
	LVDT2	FFD058	1111 1111 1101 0000 0101 1000
6	MOTOR1	FFD060	1111 1111 1101 0000 0110 0000
	LVDT1	FFD062	1111 1111 1101 0000 0110 0010
	MOTOR2	FFD064	1111 1111 1101 0000 0110 0100
	LVDT2	FFD068	1111 1111 1101 0000 0110 1000

Additional decoded control lines:

LDS - VMEbus lower data strobe
UDS - VMEbus upper data strobe
WRITE - Write/Read select
BUSY1 - LVDT-to-Digital converter 1 Busy signal
BUSY2 - LVDT-to-Digital converter 2 Busy signal

APPENDIX 2

SAMPLE EPLD DESIGN EQUATIONS

Steven Kent
ITG - FA18 Model Actuator Decoder (Card 1)
30/7/1991
1.00
A
EP900
Address Decoder

OPTIONS: SECURITY = OFF
PART: EP900
INPUTS: AS@34, A23@38, A22@37, A21@36, A20@35, A19@33, A18@32, A17@31, A16@30, A15@29, A14@13, A13@14, A12@15, A11@16, A10@17, A9@18, A8@19, A7@28, A6@27, A5@26, A4@25, A3@24, A2@23, A1@22, LDS@3, UDS@4, WRITE@5, BUSY1@11, BUSY2@12
OUTPUTS: MOTOR1@7, MOTOR2@8, LVDT1@9, LVDT2@10, DTACK@6

NETWORK:
AS = INP(AS)
A23 = INP(A23)
A22 = INP(A22)
A21 = INP(A21)
A20 = INP(A20)
A19 = INP(A19)
A18 = INP(A18)
A17 = INP(A17)
A16 = INP(A16)
A15 = INP(A15)
A14 = INP(A14)
A13 = INP(A13)
A12 = INP(A12)
A11 = INP(A11)
A10 = INP(A10)
A9 = INP(A9)
A8 = INP(A8)
A7 = INP(A7)
A6 = INP(A6)
A5 = INP(A5)
A4 = INP(A4)
A3 = INP(A3)
A2 = INP(A2)
A1 = INP(A1)
LDS = INP(LDS)
UDS = INP(UDS)
WRITE = INP(WRITE)
BUSY1 = INP(BUSY1)
BUSY2 = INP(BUSY2)
MOTOR1,MOTOR1 = COIF(MOTOR1c,Oe)
MOTOR2,MOTOR2 = COIF(MOTOR2c,Oe)
LVDT1,LVDT1 = COIF(LVDT1c,Oe)
LVDT2,LVDT2 = COIF(LVDT2c,Oe)
DTACK,DTACK = COIF(DTACKc,Oe)

```
EQUATIONS:
MOT1 =
/AS*A23*A22*A21*A20*A19*A18*A17*A16*A15*A14*/A13*A12*/A11*/A10*/A9*/A
8*/A7*/A6*/A5*A4*/A3*/A2*/A1*/LDS*/WRITE;
MOT2 =
/AS*A23*A22*A21*A20*A19*A18*A17*A16*A15*A14*/A13*A12*/A11*/A10*/A9*/A
8*/A7*/A6*/A5*A4*/A3*A2*/A1*/LDS*/WRITE;
LV1 =
/AS*A23*A22*A21*A20*A19*A18*A17*A16*A15*A14*/A13*A12*/A11*/A10*/A9*/A
8*/A7*/A6*/A5*A4*/A3*/A2*A1*/LDS*/UDS*WRITE*/BUSY1;
LV2 =
/AS*A23*A22*A21*A20*A19*A18*A17*A16*A15*A14*/A13*A12*/A11*/A10*/A9*/A
8*/A7*/A6*/A5*A4*A3*/A2*/A1*/LDS*/UDS*WRITE*/BUSY2;
DTACKc = MOT1+MOT2+LV1+LV2;
MOTOR1c = /MOT1;
MOTOR2c = /MOT2;
LVDT1c = /LV1;
LVDT2c = /LV2;
Oe = VCC;

END$
```

APPENDIX 3

MODEL ACTUATOR MODULE: BPI VECTOR USAGE

VECTOR	ADDRESS	LABEL	USE
vec96	8B60	strin1	string1 - error codes (codes, NULL)
vec97	8B61	bpiwrt	- unused
vec98	8B62	strin2	string2 - module identification string
vec99	8B63	trig	master trigger (not used in this module)
vec100	8B64	strin3	string3 - unused
vec101	8B65	clster	clear status/error buffers
vec102	8B66	strin4	string4 - unused
vec103	8B67	clerfg	clear error and flag bits
vec104	8B68	strin5	string5 - unused
vec105	8B69	travel	trigger actuator movement
vec106	8B6A	strin6	string6 - unused
vec107	8B6B	bpiwrt	- unused
vec108	8B6C	strin7	string7 - unused
vec109	8B6D	bpiwrt	- unused
vec110	8B6E	strin8	string8 - unused
vec111	8B6F	bpiwrt	- unused
vec112	8B70	rdat1	Ch1. LVDT reading to master
vec113	8B71	wdat1	Ch.1 LVDT reading to move to from master
vec114	8B72	rdat2	- unused
vec115	8B73	wdat2	Ch.1 upper limit from master
vec116	8B74	rdat3	Ch.1 current upper limit to master
vec117	8B75	wdat3	Ch.1 lower limit from master
vec118	8B76	rdat4	Ch.1 current lower limit to master
vec119	8B77	wdat4	- unused
vec120	8B78	rdat5	Ch.2 LVDT reading to master
vec121	8B79	wdat5	Ch.2 LVDT reading to move to from master
vec122	8B7A	rdat6	- unused
vec123	8B7B	wdat6	Ch.2 upper limit from master
vec124	8B7C	rdat7	Ch.2 current upper limit to master
vec125	8B7D	wdat7	Ch.2 lower limit from master
vec126	8B7E	rdat8	Ch.2 current lower limit to master
vec127	8B7F	wdat8	- unused
vec128	8B80	rdat9	Ch.3 LVDT reading to master
vec129	8B81	wdat9	Ch.3 LVDT reading to move to from master
vec130	8B82	rdat10	- unused
vec131	8B83	wdat10	Ch.3 upper limit from master
vec132	8B84	rdat11	Ch.3 current upper limit to master
vec133	8B85	wdat11	Ch.3 lower limit from master
vec134	8B86	rdat12	Ch.3 current lower limit to master
vec135	8B87	wdat12	- unused
vec136	8B88	rdat13	Ch.4 LVDT reading to master
vec137	8B89	wdat13	Ch.4 LVDT reading to move to from master
vec138	8B8A	rdat14	- unused
vec139	8B8B	wdat14	Ch.4 upper limit from master
vec140	8B8C	rdat15	Ch.4 current upper limit to master
vec141	8B8D	wdat15	Ch.4 lower limit from master
vec142	8B8E	rdat16	Ch.4 current lower limit to master
vec143	8B8F	wdat16	- unused

vec144	8B90	rdat17	Ch.5 LVDT reading to master
vec145	8B91	wdat17	Ch.5 LVDT reading to move to from master
vec146	8B92	rdat18	- unused
vec147	8B93	wdat18	Ch.5 upper limit from master
vec148	8B94	rdat19	Ch.5 current upper limit to master
vec149	8B95	wdat19	Ch.5 lower limit from master
vec150	8B96	rdat20	Ch.5 current lower limit to master
vec151	8B97	wdat20	- unused
vec152	8B98	rdat21	Ch.6 LVDT reading to master
vec153	8B99	wdat21	Ch.6 LVDT reading to move to from master
vec154	8B9A	rdat22	- unused
vec155	8B9B	wdat22	Ch.6 upper limit from master
vec156	8B9C	rdat23	Ch.6 current upper limit to master
vec157	8B9D	wdat23	Ch.6 lower limit from master
vec158	8B9E	rdat24	Ch.6 current lower limit to master
vec159	8B9F	wdat24	- unused
vec160	8BA0	rdat25	Ch.7 LVDT reading to master
vec161	8BA1	wdat25	Ch.7 LVDT reading to move to from master
vec162	8BA2	rdat26	- unused
vec163	8BA3	wdat26	Ch.7 upper limit from master
vec164	8BA4	rdat27	Ch.7 current upper limit to master
vec165	8BA5	wdat27	Ch.7 lower limit from master
vec166	8BA6	rdat28	Ch.7 current lower limit to master
vec167	8BA7	wdat28	- unused
vec168	8BA8	rdat29	Ch.8 LVDT reading to master
vec169	8BA9	wdat29	Ch.8 LVDT reading to move to from master
vec170	8BAA	rdat30	- unused
vec171	8BAB	wdat30	Ch.8 upper limit from master
vec172	8BAC	rdat31	Ch.8 current upper limit to master
vec173	8BAD	wdat31	Ch.8 lower limit from master
vec174	8BAE	rdat32	Ch.8 current lower limit to master
vec175	8BAF	wdat32	- unused
vec176	8BB0	rdat33	Ch.9 LVDT reading to master
vec177	8BB1	wdat33	Ch.9 LVDT reading to move to from master
vec178	8BB2	rdat34	- unused
vec179	8BB3	wdat34	Ch.9 upper limit from master
vec180	8BB4	rdat35	Ch.9 current upper limit to master
vec181	8BB5	wdat35	Ch.9 lower limit from master
vec182	8BB6	rdat36	Ch.9 current lower limit to master
vec183	8BB7	wdat36	- unused
vec184	8BB8	rdat37	Ch.10 LVDT reading to master
vec185	8BB9	wdat37	Ch.10 LVDT reading to move to from master
vec186	8BBA	rdat38	- unused
vec187	8BBB	wdat38	Ch.10 upper limit from master
vec188	8BBC	rdat39	Ch.10 current upper limit to master
vec189	8BBD	wdat39	Ch.10 lower limit from master
vec190	8BBE	rdat40	Ch.10 current lower limit to master
vec191	8BBF	wdat40	- unused
vec192	8BC0	rdat41	Ch.11 LVDT reading to master
vec193	8BC1	wdat41	Ch.11 LVDT reading to move to from master
vec194	8BC2	rdat42	- unused
vec195	8BC3	wdat42	Ch.11 upper limit from master
vec196	8BC4	rdat43	Ch.11 current upper limit to master
vec197	8BC5	wdat43	Ch.11 lower limit from master
vec198	8BC6	rdat44	Ch.11 current lower limit to master
vec199	8BC7	wdat44	- unused

vec200	8BC8	rdat45	Ch.12 LVDT reading to master
vec201	8BC9	wdat45	Ch.12 LVDT reading to move to from master
vec202	8BCA	rdat46	- unused
vec203	8BCB	wdat46	Ch.12 upper limit from master
vec204	8BCC	rdat47	Ch.12 current upper limit to master
vec205	8BCD	wdat47	Ch.12 lower limit from master
vec206	8BCE	rdat48	Ch.12 current lower limit to master
vec207	8BCF	wdat48	- unused
vec208	8BD0	rdst1	motor status register
vec209	8BD1	testmode	enter motor manual drive mode
vec210	8BD2	rdst2	- unused
vec211	8BD3	direct	direction for motor to move
vec212	8BD4	rdst3	- unused
vec213	8BD5	teststop	exit motor manual test mode
vec214	8BD6	rdst4	- unused
vec215	8BD7	lvdtonly	LVDT read only channel vector
vec216	8BD8	rdst5	- unused
vec217	8BD9	power	motor power relay control
vec218	8BDA	rdst6	- unused
vec219	8BDB	onpulse	turn manual pulse mode on
vec220	8BDC	rdst7	- unused
vec221	8BDD	offpulse	turn manual pulse mode off
.	.	.	
.	.	unused	
.	.	.	
vec255	8BFF		

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central

Chief Defence Scientist
AS, Science Corporate Management } shared copy
FAS Science Policy
Counsellor, Defence Science, London (Doc Data sheet only)
Counsellor, Defence Science, Washington (Doc Data sheet only)
S.A. to Thailand MRD (Doc Data sheet only)
S.A. to the DRC (Kuala Lumpur) (Doc Data sheet only)
Scientific Adviser, Defence Central
OIC TRS, Defence Central Library
Document Exchange Centre, DSTIC (8 copies)
Defence Intelligence Organisation
Librarian, Defence Signals Directorate, (Doc Data sheet only)
Director General - Army Development (NSO) (4 copies)
Industry Policy and Programs Branch, FAS

Aeronautical Research Laboratory

Director
Library
Chief Air Operations Division
Author: S.A. Kent (2 copies)
C. Martin
S. Lam
N. Matheson
J. Harvey

Materials Research Laboratory

Director/Library

Defence Science & Technology Organisation - Salisbury

Library

Navy Office

Navy Scientific Adviser (3 copies Doc Data sheet only)
Aircraft Maintenance and Flight Trials Unit

Army Office

Scientific Adviser - Army (Doc Data sheet only)
Engineering Development Establishment Library

Air Force Office

Air Force Scientific Adviser
Aircraft Research and Development Unit
Scientific Flight Group
Library
PDR AF
DGELS AIRREG4 HQLC
DCFD(AIR)
OIC ATF, ATS, RAAFSTT, WAGGA (2 copies)

Other Organisations

NASA (Canberra)
AGPS

SPARES (6 COPIES)

TOTAL (47 COPIES)

AL 149 DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-006-658	ARL-TN-13	JUNE 1993	ADA 90/061

4. TITLE

A WIND TUNNEL MODEL CONTROL SURFACE ACTUATOR INTERFACE

5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF. (C) RESTRICTED (R), LIMITED (L), UNCLASSIFIED (U)).

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES
39

7. NO. REFS.
2

8. AUTHOR(S)

S.A. KENT

9. DOWNGRADING/DELIMITING INSTRUCTIONS

Not applicable.

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY

AIR OPERATIONS DIVISION

506 LORIMER STREET

FISHERMENS BEND VIC 3207

11. OFFICE/POSITION RESPONSIBLE FOR:

SPONSOR RAAF DGFD (AIR) DCFD

SECURITY -

DOWNGRADING -

APPROVAL CAOD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)

Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO ....

No limitations.

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE [X] UNRESTRICTED OR [ ] AS FOR 13a.

14. DESCRIPTORS

Wind tunnel models
Aircraft models
Control surfaces
Remotely actuated control system
F/A-18 aircraft

15. DISCAT SUBJECT CATEGORIES
010101

16. ABSTRACT

A microprocessor controlled system is described for the remote positioning of flight control surfaces on a wind tunnel aircraft model. The system utilises DC micromotors and Linear Variable Displacement Transducers (LVDTs) for driving force and accurate position feedback. The Actuator Module was developed primarily for use with a 1/9th scale F/A-18 model to collect data for the International Follow On Structural Test Program (IFOSTP).

PAGE CLASSIFICATION
UNCLASSIFIED
PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT).

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER	19. WA NUMBER	20. TYPE OF REPORT AND PERIOD COVERED
Technical Note 13	76 700B	

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)